Bag-of-Audio-Words Approach for Multimedia Event Classification

Stephanie Pancoast[1,2], *Murat Akbacak*[1]

[1] Speech Technology and Research Lab, SRI International, Menlo Park, CA
[2] Department of Electrical Engineering, Stanford University, Stanford, CA

Abstract

With the popularity of online multimedia videos, there has been much interest in recent years in acoustic event detection and classification for the improvement of online video search. The audio component of a video has the potential to contribute significantly to multimedia event classification. Recent research in audio document classification has drawn parallels to text and image document retrieval by employing what is referred to as the bag-of-audio words (BoAW) method. Compared to supervised approaches where audio concept detectors are trained using annotated data and extracted labels are used as low-level features for multimedia event classification. The BoAW approach extracts audio concepts in an unsupervised fashion. Hence this method has the advantage that it can be employed easily for a new set of audio concepts in multimedia videos without going through a laborious annotation effort. In this paper, we explore variations of the BoAW method and present results on NIST 2011 multimedia event detection (MED) dataset.

Index Terms: Bag-of-audio-words, multimedia event detection

1. Introduction

Because of the popularity of online videos, there has been much interest in recent years in multimedia analysis. More specifically, multimedia event detection (MED) tasks require a system that can search user-submitted quality videos for specific events [1]. Features in the video imagery play a significant role in determining the content. However, the audio component for a given video can also be critical. Consider the case of detecting a home run in a baseball game. From the still frames, it may be determined that the setting is a baseball game or players are in action on the field, but without the capability to detect cheering in the audio, it would be even more difficult to discriminate between an uneventful game and one with a home run.

Recent research has explored various techniques for modeling the audio component to better understand the multimedia content. These techniques can be grouped in two main categories. The first group of methods require annotations of acoustic concepts so that detectors can be trained in a supervised fashion as is done in recent works. Later these detectors are used to extract acoustic concept labels from the multimedia document to be used as low-level features in multimedia event classification. This approach has the advantage of having human interpretable labels which becomes important if users want to search for specific concepts in a multimedia retrieval task using text query. On the other hand, manually determining the list of audio concepts requires expert knowledge and this process is laborious for large datasets, especially when the annotation work has to be repeated for a new set of acoustic concepts appearing in new multimedia events. The second group of methods does not require manually defined labels and annotated data for acoustic events or concepts. Instead, the concept labels are created in an unsupervised way from the actual data. The method employed in this work falls under this category.

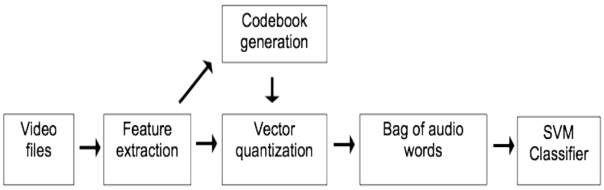

Figure 1: Diagram of the Bag-of-Audio-Words pipeline.

Similar to well established techniques for classifying text documents (*bag-of-words*) and image documents (*bag-of-visual-words*) [2], we model the audio component of multimedia videos using the *bag-of-audio-words* method. The BoAW has also recently been used for both audio document retrieval [3], copy detection [4], and MED tasks [5].

As illustrated in Figure 1, the BoAW method involves generating "words" with a clustering algorithm, quantizing the original features to generate the "bag-of-words" in the form of a histogram, and performing the classification task on the histograms. There are numerous variations at different steps of the BoAW algorithm including differing choices for front-end features, codebook sizes for generating the codewords, histogram normalization techniques, and classifiers. A study of these variations for image scene classification found that choices in the algorithm that are optimal for bag-of-words are not necessarily optimal for bag-of-visual-words [2]. While BoAW is especially similar to bag-of-visual-words, in the sense that "words" are artificially created via clustering, the problem differs fundamentally in the nature of original feature space. The features extracted from images are generally scale-invariant. Each feature describes a component of the image such as an edge or a color. For audio documents, the features are extracted from the one-dimensional signal at fixed length intervals. These intervals, however, may not capture the full acoustic variation that characterizes a given sound. Because of these differences, it is a natural extension to consider that certain representation choices in the bag-of-visual-words algorithm are not optimal when representing audio.

In this paper we discuss our dataset and experimental setup in Sections 2 and 3 respectively. In Section 4 we then describe the BoAW algorithm and its variations as previously mentioned in detail. In Section 5, we present results for each of these representation choices on the NIST TRECVID 2011 DEV-T dataset.

2. Dataset

We the National Institute of Standards and Technology (NIST) development data provided for the Text Retrieval Conferences

Event Name	# Segments in DEV-T
Attempting a board trick	114
Feeding an animal	114
Landing a fish	86
Wedding ceremony	89
Working on a woodworking project	100

Table 1: Event class names and number of segments in the DEV-T test set.

Video Retrieval Evaluation (TRECVID) 2011 multimedia event detection track [1]. We used 2062 videos from the events kit for training and 4292 from the Transparent Development (DEV-T) collection for testing. Each file of the events kit is labeled with one of fifteen video events ranging from birthday parties to animal grooming. The DEV-T set contains only the first five video events. These are listed in Table 1 along with the number of segments for each event. Of the DEV-T files, 3789 are labeled as "none," indicating that these videos should be classified as negative samples for all verification experiments. The videos were provided in MP4 format with an audio sampling rate at 16 kHz.

3. Task Setup

We chose to perform verification, also referred to as one-against-all, experiments. For each of the five experiments, a given file is labeled as *in-class* or *out-of-class*. Examples include wedding vs. not wedding and board trick vs. not board trick. This allows us to approach the problem as a set of binary classifications. If the different models performed well, the likelihood scores for each verification experiment could be combined to make a final decision on which of the five video events that file belonged to.

4. Bag-of-Audio-Words and Model Choices

Documents, whether written, visual, or audio, vary in length. Features representing these documents are often not fixed-length and as a result cannot be used directly with many classification techniques. Representing the document as a "bag-of-words" resolves this issue by representing the variable-length file with a fixed-length histogram, also referred to as a word-vector. As mentioned, a diagram showing the bag-of-words system is presented in Figure 1. When the words do not exist naturally like with text documents, a codebook is created using a clustering algorithm. The Lloyd (k-means) clustering is a common choice for this step. The centroids of the resulting clusters are taken as the codewords, and the original feature vectors are replaced by a single index representing the nearest codeword to that original vector. This process is called vector quantization [6]. The "bag" is then created by simply generating a histogram of codewords in the given file. At this point every document is represented by a fixed-length histogram and can be passed to the classifier to complete the system.

Numerous model parameters and modifications have been explored for written and image document classification to improve this basic system. The remainder of this section describes model choices applicable to audio document classification, and results of these modifications on our dataset are presented in Section 5.

4.1. Front-end Features

The mel frequency cepstrum represents the short-term power spectrum of a sound. It is based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency. The cepstrum can be thought of as a "spectrum of the spectrum" and provides information on how the spectrum energy changes over time. The MFCCs are the coefficients that make up the mel cepstrum [7]. The mel scale on which the cesptrum is computed captures the nonlinearity of human hearing.

MFCCs are standard features used in many speech recognition tasks such as automatic speech recognition and speaker identification. The features used in our experiments are computed for every 10 ms audio segment and are extracted using a hamming window with 50% overlap. The features consist of 12 MFCCs as well as the log energy. The first and second derivates of each coefficient as well as the log energy are concatenated with the original features to result in a 39-dimensional feature vector.

4.2. Codebook Size

To generate the "words," we must determine the number of codewords the clustering algorithm should generate. This model parameter is referred to as the codebook size, and its value is a tradeoff. When generating a small set of clusters, dissimilar sounds (as modeled by the MFCC vector) can be grouped into the same audio-word, and the codebook is therefore more general but not necessarily discriminative. On the other hand, larger codebook sizes assign similar sounds to different audio-words, resulting in a more discriminative but less general vocabulary. Computation time to generate the codewords also increases with larger codebook sizes. We explored codebook sizes ranging from 500 to 2000 and the impact of this parameter on the final MED accuracy.

4.3. Word Vector Representation

As mentioned previously, a document is represented by a "bag" of words. This bag is called a histogram, or word vector, and each element represents the count of occurrences of a given "word" in the document. Histogram normalization is a common step between quantization and classification. Because the documents vary in length, histograms representing a longer document will have overall higher counts. To eliminate the influence of document length, the L1 normalization, also referred to as term frequency normalization, is used [2]. A binary vector is also occasionally used to represent the document.

4.4. Classifier

Once the documents are each represented as a bag of "words", we use a learning model to perform the classification with the histograms as input features. We chose support vector machines (SVMs) due their ability to model nonlinear decision boundaries using what is referred to as the 'kernel trick'. SVMs use a kernel function to compute the inner product between feature vectors. The optimal kernel therefore depends on the nature of the input features. For bag-of-words related classification tasks, the linear, radial basis function (RBF), and histogram intersection kernel are commonly used.

The linear kernel is popular due to its simplicity; however, it ignores the nature of the input feature space. The RBF kernel performs well for a large range of applications and is often used when little is known about the input feature space [2]. The histogram intersection kernel is optimal for comparing histograms

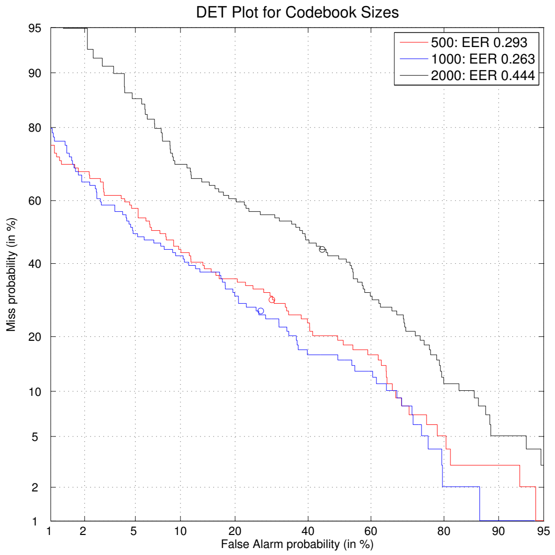

Figure 2: DET Curve for different codebook sizes using fixed kernel and no histogram normalization. Results compiled across all five verification experiments.

and is used for image classification based on color distribution [8]. This kernel efficiently computes the inner product while also capturing the property that each similar histogram will have overlapping bins. For two feature vectors, **x**, **y** $\in R^p$, the inner product is calculated by

$$k(\mathbf{x},\mathbf{y}) = \mathbf{x}^T\mathbf{y} + c \tag{1}$$

The radial basis function kernel is calculated as

$$k(\mathbf{x},\mathbf{y}) = exp(-\gamma\|\mathbf{x}-\mathbf{y}\|^2) \tag{2}$$

where γ is the default value of $1/p$. The histogram intersection kernel calculates the inner product by

$$k(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{p} min(\mathbf{x}_i,\mathbf{y}_i) \tag{3}$$

5. Results and Analysis

In this section we present results from variations of the BoAW method on our dataset. Results are presented in the form of Detection Error Tradeoff (DET) curves which are commonly used to show the tradeoff between false alarm errors and missed detection errors. We generating the DET-curves in this paper with plotting software available from the NIST website [9].

5.1. Codebook Size

We explored codebook sizes in the range previously used for the BoAW methods. Results are presented in Figure 2. The DET curve shows that the larger codebook size of 2000 codewords clearly performs worse than smaller codebook sizes. The 1000-word codebook slightly outperforms the BoAW algorithm using 500 codewords, so we used this value for the remainder of the experiments.

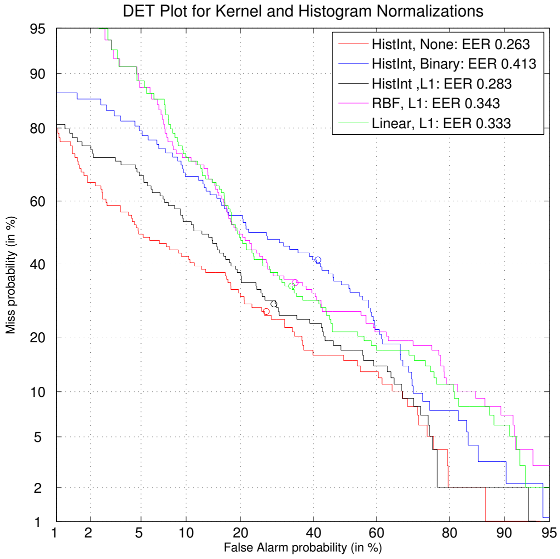

Figure 3: DET Curve for different kernel and histogram normalization techniques. Results compiled across all five verification experiments.

5.2. Histogram Normalizations and SVM Kernels

We ran the BoAW experiments with a fixed codebook size of 1000 and varied the normalization procedure applied to the histogram before classification as well as the kernel used in the SVM classifier. Results from these experiments are presented in Figure 3. The curves show the performance for various histogram, kernel combinations collated across all verification experiments.

The RBF performs poorly on unnormalized histograms, so we present the results using RBF and linear kernels when applied to L1-normalized histograms. The binary representation of the BoAW, however, still performs poorly as expected, since much of the information about the content of the video is lost when reducing the elements to zeros and ones. The DET curve shows the histogram intersection kernel based experiments clearly outperforming those using the linear and RBF kernels for the L1 and unnormalized histograms. This is not surprising as the histogram intersection kernel is optimized for histogram comparison as we have here. It is interesting however that the L1 normalization performs worse than the unnormalized histograms, suggesting that the length of the video is somewhat correlated with the video event class.

5.3. Performance by Event

We used the best-performing model parameters from earlier experiments (codebook size 1000, no histogram normalization, and histogram intersection kernel) to observe how results vary for each of the five video event classes. Results are presented in Figure 4. The woodworking project performs the best across all events. The wedding ceremony videos perform well in the low false alarm region. The other three video are still an improvement on a random decision, but do not see performance as high as with the previously two mentioned events. These observations are likely due to the typical video properties for each event. Woodworking projects largely contain the same type of

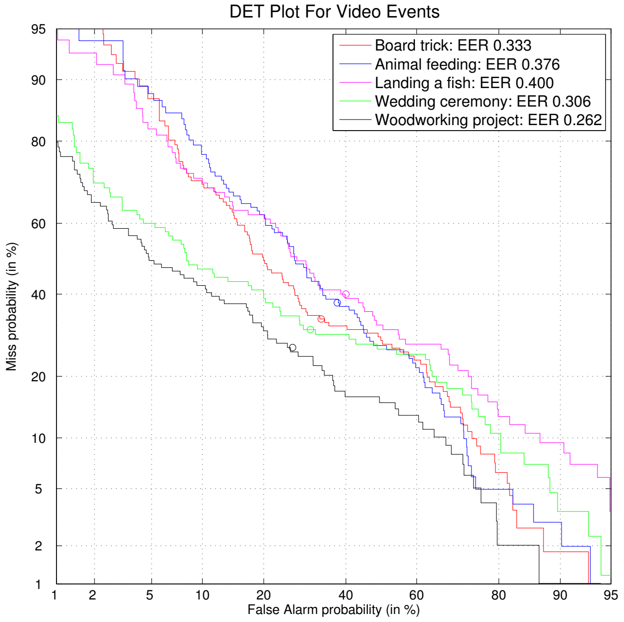

Figure 4: DET Curve for each of the video event classes using the histogram intersection kernel codebook size 1000 and no histogram normalization.

tool sounds throughout the video, which would result in histograms with a few large-valued bins. Wedding ceremonies, similarly, are generally high quality and quiet with occasional voices, music, or wind sounds present. Animal feeding, board tricks, and landing a fish, however, are typically low-quality videos with a broader range of sounds scattered throughout the file.

6. Conclusion

In this paper we presented an analysis of the BoAW method applied to a multimedia event detection task in the form of verification experiments on five video event classes. Results from the experiments show that common variations of this method used for text and video document modeling such as L1-normalization are not necessarily the optimal choice when applying the method to audio documents. The histogram intersection kernel with no histogram normalization and codebook size 1000 showed the best performance on the MED verification tasks. It was also found that results depend on the acoustic variation of the video.

Future work will explore further enhancements on the bag-of-audio-words technique. Other front-end features used in speech-related research such as perceptual linear prediction features may enhance performance. Also, in these experiments we assumed that the "words" were fixed length and evenly spread throughout the document. As mentioned is Section 1, this is a valid assumption for bag-of-visual-words as the original feature space is often extracted independently from the size, orientation, and placement of the object that feature is describing. For audio documents, the true "word" may be variable length. Future work will extend the BoAW technique to account for this property. Once optimized, the BoAW system results can be combined with other audio and video processing systems to enhance performance on multimedia event detection tasks.

7. Acknowledgments

We thank Keon van de Sande, Ramesh Nallapati, Eric Yeh, and Professor Robert M. Gray for their valuable discussions. This work was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center contract number D11PC0067. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes nonwithstanding any copyright annotation thereon.

Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.

8. References

[1] "TRECVID multimedia event detection 2011 evaluation," *http://www.nist.gov/itl/iad/mig/med11.cfm*

[2] Yang, J., Jiang, Y., Hauptmann, A., and Ngo, C., "Evaluating bag-of-visual-words representations in scene classification," in *Proceedings of the International Workshop on Multimedia Information Retrieval*, pp. 197-206, 2007.

[3] Checkik, G., Le, E., Rehn, M., Bengio, S., and Lyon, D., *Large-scale Content-Based Audio Retrieval from Text Queries*, 2008.

[4] Uchida, Y., Sakazawa, S. Agrawal, M., Akbacak, M., "KDDI Labs and SRI International at TRECVID 2010: Content-Based Copy Detection", in *NIST TRECVID 2010 Evaluation Workshop*, November 2010, Gaithersburd, MD.

[5] Jiang, Y., Zang, X., Ye, G., Bhattacharya, S., Ellis, D. Shah, M., and Chang, S., "Columbia-UCF TRECVID2010 Multimedia Event Detection: Combining Multiple Modalities, Contextual Concepts, and Temporal Matching," 2001.

[6] Gersho, A., and Gray, R., *Vector Quantization and Signal Compression*, Boston: Kluwer Academic Publishers, 1992.

[7] Rabiner, L. R., and Schafer, R. W., *Introduction to Digital Speech Processing*, 2007.

[8] Barla, A., Odone, F., and Verri, A., "Histogram Intersection Kernel for Image Classification," in *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, 2003.

[9] NIST DETware V.2. [Online] Available: http://www.itl.nist.gov/iad/mig/tools/.